eSecurity

Treinamentos em Segurança da Informação

O que temos para hoje?

Menu do dia:

- ✓ **O protocolo HTTP**
  - ✓ Entendendo os Métodos
  - ✓ Métodos Get, Head, Post, Put, Delete, Trace, Options, Connect
- ✓ **Varreduras Intrusivas**
  - ✓ Conhecendo o Nmap
    - ✓ Buscando alvos por Methods
    - ✓ Respostas Trace, Delete, Put e Connect
    - ✓ Enumeração de Subpastas
    - ✓ Efetuando Brute Force em HTTPAuth
    - ✓ Efetuando Brute Force em Formulários

O protocolo HTTP - Conceito

O Hypertext Transfer Protocol (HTTP), em português Protocolo de Transferência de Hipertexto, é um protocolo de comunicação (na camada de aplicação segundo o Modelo OSI) utilizado para sistemas de informação de hipermídia, distribuídos e colaborativos. Ele é a base para a comunicação de dados da World Wide Web.

Hipertexto é o texto estruturado que utiliza ligações lógicas (hiperlinks) entre nós contendo texto. O HTTP é o protocolo para a troca ou transferência de hipertexto.

Uma sessão HTTP é uma sequência de transações de rede de requisição-resposta. Um cliente HTTP inicia uma requisição estabelecendo uma conexão Transmission Control Protocol (TCP) para uma porta particular de um servidor (normalmente a porta 80).
Um servidor HTTP ouvindo naquela porta espera por uma mensagem de requisição de cliente. Recebendo a requisição, o servidor retorna uma linha de estado, como "HTTP/1.1 200 OK", e uma mensagem particular própria. O corpo desta mensagem normalmente é o recurso solicitado, apesar de uma mensagem de erro ou outra informação também poder ser retornada.

O protocolo HTTP - Métodos

O protocolo HTTP define oito métodos (GET, HEAD, POST, PUT, DELETE, TRACE, OPTIONS e CONNECT) que indicam a ação a ser realizada no recurso especificado.
O método determina o que o servidor deve fazer com o URL fornecido no momento da requisição de um recurso. Um servidor HTTP deve implementar ao menos os métodos GET e HEAD.

Método GET
Solicita algum recurso como um arquivo ou um script CGI (qualquer dado que estiver identificado pelo URI) por meio do protocolo HTTP.
Por exemplo, segue abaixo uma comunicação entre um cliente e um servidor HTTP. O servidor possui a URL www.exemplo.com, porta 80.

O pedido do cliente:

```
GET /index.html HTTP/1.1
Host: www.exemplo.com
```

O protocolo HTTP - Métodos

O cabeçalho Host reconhece vários diferentes nomes DNS que tenham o mesmo IP.

A resposta do servidor:

HTTP/1.1 200 OK
Date: Mon, 23 May 2005 22:38:34 GMT
Server: Apache/1.3.27 (Unix) (Red-Hat/Linux)
Last-Modified: Wed, 08 Jan 2003 23:11:55 GMT
Etag: "3f80f-1b6-3e1cb03b"
Accept-Ranges: bytes
Content-Length: 438
Connection: close
Content-Type: text/html; charset=UTF-8

O protocolo HTTP - Métodos

Método HEAD
Variação do GET em que o recurso não é retornado. É usado para obter informações por meio do cabeçalho da resposta, sem ter que recuperar todo o conteúdo.

Método POST
Envia dados para serem processados (por exemplo, dados de um formulário HTML) para o recurso especificado. Os dados são incluídos no corpo do comando. Sua utilização em uma requisição ocorre quando é necessário enviar dados ao servidor para serem processados, geralmente por um programa script identificado no Request-URI. Uma requisição por meio desse método sempre requer que as informações submetidas sejam incluídas no corpo da mensagem e formatadas como uma query string, além de conter cabeçalhos adicionais especificando seu tamanho (Content-Length) e seu formato (Content-Type). Por isso, esse método oferece uma maior segurança em relação aos dados transferidos, ao contrário do método GET que os dados são anexados a URL, ficando visíveis ao usuário.

O protocolo HTTP - Métodos

Por exemplo:

```
POST /index.html HTTP/1.0
Accept: text/html
If-modified-since: Sat, 29 Oct 1999 19:43:31 GMT
Content-Type: application/x-www-form-urlencoded
Content-Length: 41
Nome=NomePessoa&Idade=99&Curso=Computacao
```

Método PUT
Envia certo recurso

Método Delete
Exclui o recurso.

Método Trace
Ecoa o pedido, de maneira que o cliente possa saber o que os servidores intermediários estão mudando em seu pedido.

O protocolo HTTP - Métodos

Método Options
Recupera os métodos HTTP que o servidor aceita.

Método Connect
Serve para uso com um proxy que possa se tornar um túnel SSL (um túnel pode ser usado, por exemplo, para criar uma conexão segura).

Fonte: Wikipedia

Conhecendo o Nmap

Varreduras Intrusivas: Nmap

Servidores Web suportam diferentes métodos HTTP de acordo com sua configuração e software, alguns deles podem ser perigosos em certas condições.

Pentesters precisam encontrar uma maneira listar rapidamente os métodos disponíveis. O script NSE *http-methods* permite não só listar estes métodos potencialmente perigosos, mas também para testá-los.

Este exemplo mostra como usar o Nmap para enumerar todos os métodos HTTP suportados por um servidor web.

nmap -p80,443 --script http-methods scanme.nmap.org

Para verificar individualmente a resposta do status dos métodos retornados por OPÇÕES, usar o script argumentativo http-methods.retest:

nmap -p80,443 --script http-methods --script-args http-methods.retest scanme.nmap.org

Varreduras Intrusivas: Nmap

Por padrão, o script http-métodos usa a pasta raiz como o caminho base (/). Se você pretende definir um caminho de base diferente, defina o argumento http-methods.url-path:

nmap -p80,443 --script http-methods --script-args http-methods.urlpath=/teste/ scanme.nmap.org

HTTP methods TRACE, CONNECT, PUT e DELETE podem apresentar um risco de segurança, e eles precisam ser testados exaustivamente caso sejam suportado por um servidor web ou aplicação.

TRACE faz aplicações suscetíveis a ataques Cross Site Tracing (XST) e pode fazer com que os atacantes acessem os cookies marcados como httpOnly.

O CONNECT method pode permitir que o servidor web para ser usado como um proxy web não autorizado.

Os métodos PUT e DELETE têm a capacidade de alterar o conteúdo de uma pasta, e isso pode, obviamente, ser utilizado se as permissões não estão definidas corretamente.

Varreduras Intrusivas: Nmap

Você pode enumerar subpastas dentro de determinadas na raiz do servidor com o argumento basepath:

nmap --script http-enum http-enum.basepath=/web/ -p80 192.168.2.104

Para acompanhar um status mais detalhado sobre o retorno da enumeração, você pode utilizar a opção enum.displayall:

nmap --script http-enum http-enum.displayall -p80 192.168.2.104

Varreduras Intrusivas: Nmap

Muitos roteadores domésticos, webcams IP, e até mesmo aplicações web ainda dependem de autenticação HTTP nos dias de hoje, e pentesters precisam utilizar uma lista de palavras de senhas para certificar se o sistema ou contas de usuário são seguros.

Agora, graças ao script NSE http-brute, podemos efetuar testes robustos baseados em dicionários contra recursos HTTPAuth protegidas.

Esta dica mostra como realizar auditoria senha de força bruta contra servidores web que estão usando a autenticação HTTP.

Use o seguinte comando para executar a auditoria do Nmap brute force contra um recurso protegido pela autenticação básica de HTTP:

```
nmap -p80 --script http-brute --script-args userdb=arquivo.txt,passdb=arquivo.txt www.laboratoriohacker.com.br -d
```

Varreduras Intrusivas: Nmap

Agora vamos supor que a pasta ao qual necessita senha não seja a pasta principal e sim uma subpasta, tal como /admin.

Vejamos como ficaria o comando:

```
nmap -p80 --script http-brute --script-args userdb=arquivo.txt,passdb=arquivo.txt
http-user.path=/admin/ www.laboratoriohacker.com.br –d
```

Quando o Nmap encontra uma credencial, ele não finaliza os testes das demais senhas em busca de novas possíveis credenciais. Para eliminar esse processo e solicitar que o Nmap encerre a varredura após encontrar a primeira credencial, você terá que utilizar a opção brute.firstOnly:

```
nmap -p80 --script http-brute --script-args userdb=arquivo.txt,passdb=arquivo.txt
brute.firstOnly http-user.path=/admin/ www.laboratoriohacker.com.br –d
```

Varreduras Intrusivas: Nmap

Agora vamos ao ataque à um formulário de autenticação. Esta opção requer um conhecimento mais aprimorado e um levantamento de dados melhor, antes de efetuar o ataque:

```
nmap -p80 --script http-form-brute --script-args http-form-brute.uservar=login,http-form-brute.passvar=senha,userdb=arquivo.txt,passdb=arquivo.txt,http-form-brute.onfailure=incorretos,http-form-brute.path=/adm/verificar.php www.laboratoriohacker.com.br -d
```

Chega por hoje

www.eSecurity.com.br

E-mail: alan.sanches@esecurity.com.br

Twitter: @esecuritybr e @desafiohacker

Skype: desafiohacker

Fanpage: www.facebook.com/academiahacker